

Маковський Д. Ю.

студент фізико-математичного факультету

Усата О.Ю.

доцент, кандидат педагогічних наук,

доцент кафедри прикладної математики та інформатики,

Житомирський державний університет імені Івана Франка

РОЗРОБКА КРОС-ПЛАТФОРМНИХ ДОДАТКІВ ДЛЯ МОБІЛЬНИХ ПРИСТРОЇВ

***Анотація.** З кожним днем розробка програмного забезпечення стає все більш розповсюдженим явищем. Особливо стрімкого розвитку зараз набули мобільні пристрої. В статті розглянуто головні аспекти розробки програмного забезпечення для усіх відомих мобільних платформ, на прикладі ОС Android.*

***Ключові слова.** Xamarin, крос-платформність, C#, Android.*

Аннотация. *С каждым днем разработка программного обеспечения становится все более распространенным явлением. Особенно стремительного развития сейчас получили мобильные устройства. В статье рассмотрены основные аспекты разработки программного обеспечения для всех известных мобильных платформ, на примере ОС Android.*

Ключевые слова. *Xamarin, кросс-платформенность, C#, Android.*

Annotation. *Every day, software development is becoming more common. Especially the rapid development of mobile devices now have. The article deals with the basic aspects of software development for all known mobile platforms, the example of the Android OS.*

Key words. *Xamarin, crossplatform, C#, Android.*

Впродовж останніх декількох років у розробників все частіше поставало питання реалізації програмних рішень для усіх відомих платформ. Також сюди входять такі проблеми як: неприпустимі витрати часу та коштів на розробку, неповноцінність функціоналу, повільна швидкодія. Серед мобільних пристроїв для побудови крос-платформного додатку розглянуто фреймворк Xamarin, разом з його основними властивостями, перевагами та недоліками, що будуть впливати на вирішення постановленого завдання.

Сьогодні складно здивувати сучасну людину смартфоном, планшетом, або комп'ютером. Добре це чи погано, але ці пристрої стали частиною нашого життя, і вони продовжуватимуть впливати на масову культуру й далі. Більш того, їх розробкою займаються декілька провідних конкуруючих компаній.

З точки зору розробника, таке різноманіття стає швидше перешкодою, адже практично для кожної із усіх відомих платформ існує свій набір програмного забезпечення (SDK), що, як правило, не сумісний з іншими платформами. Таким чином, розробка для кожної із платформ окремо, – дороге і складне завдання.

Розробнику, який візьметься за створення додатку для Android, IOS та Windows, необхідно володіти такими мовами програмування як Java, Objective-C (або Swift) та C#. У такому випадку одна задача породжуватиме ще цілих три

окремих продукти, які буде необхідно розробляти і їх підтримувати. Для вирішення проблеми крос-платформності допустима реалізація у вигляді HTML5-додатку, але він буде досить повільним, і не матиме доступу до системних ресурсів. Як наслідок, реалізація за допомогою веб-технологій не є кращим вибором на даний момент. Додаток повинен працювати з різними платформами не тільки в експериментальних умовах, а й на самих звичайних мобільних пристроях, подекуди не самих потужних, і з різними розмірами дисплею.

Популярність мобільних пристроїв невідомо зростає. З кожним роком їх кількість збільшується. У 86% смартфонів, що було продано у другому кварталі 2014 року встановлено операційну систему Android. При цьому за весь 2014 рік було продано більш ніж мільярд Android-пристроїв. З іншого боку, близько 40% корпоративного сегменту США використовує мобільні пристрої під керуванням IOS. А тим часом, Microsoft активно займається розробкою та випуском Windows 10 Mobile [4].

При такому різноманітті та розповсюдженні мобільних пристроїв питання крос-платформності стає саме по собі. В цьому спектрі найбільш помітним стає Xamarin, – фреймворк (комплексна бібліотека) для крос-платформної розробки мобільних додатків із використанням C#.

Метою даної статті є огляд основних характеристик та можливостей розробки крос-платформних додатків із використанням Xamarin та C#, враховуючи основні переваги та недоліки.

Команда розробників Xamarin у першу чергу намагається задовольнити усі потреби та побажання розробників (на даний момент їх близько 1,5 мільйона), розуміючи, що саме від них залежатиме, скільки буде розроблено додатків з використанням Xamarin. Це ми можемо стверджувати, зважаючи на документацію на офіційному сайті, та на кількість доповнень (більша їх частина безкоштовна), які можна завантажити та встановити у свій додаток. Для початківців, які використовують Xamarin, формат пошуку допомоги на форумах є досить незвичним, – розробник здебільшого буде отримувати

відповіді на свої питання на форумах, що присвячено темам .NET, C#, Objective-C(або Swift) для IOS, та Java із Android-SDK [2].

Ми виділили основні переваги, які, на нашу думку, є головними у аспектах вирішення поставленого завдання:

- розробка для IOS, Android та MAC з використанням C# та .Net Framework;
- можливості мови програмування C#, такі як LINQ, лямбда-вирази, Generic та async;
- можливість розробки у Visual Studio. Також присутнє рідне середовище розробки, Xamarin Studio;
- використання рідних UI (User Interface) у додатках для кожної із платформ;
- швидкодія;
- повний доступ до API обраної платформи;
- безкоштовна версія, що містить в собі достатній функціонал для повноцінної розробки;
- підтримка Google Glass.

Ми вважаємо, що Xamarin варто використовувати тоді, коли додаток необхідно реалізувати для роботи як мінімум на двох основних платформах, таких як: Android та IOS. Але пристрої, що працюють під їх керуванням різні, починаючи з апаратного забезпечення, закінчуючи логікою роботи операційної системи та ліцензуванням в цілому. Тому варто вдатися в деякі технічні аспекти, що будуть пов'язані на пряму з реалізацією крос-платформності [1].

Перш за все, фреймворк Xamarin складається з декількох основних частин:

- Xamarin.IOS – бібліотека класів для C#, що надає розробнику доступ до IOS SDK;
- Xamarin.Android – бібліотека класів для C#, що аналогічно надає розробнику доступ до Android SDK;
- компілятори для iOS та Android;

- середовище розробки Xamarin Studio;
- плагін для Visual Studio (працює і для безкоштовного випуску Community Edition).

Основою для Xamarin виступає інший фреймворк – Mono (вільна реалізація .NET). Вона включає в себе власний компілятор C#, середовище виконання, а також усі основні .NET бібліотеки. Головна мета проекту – дозволити запускати програми, що написані C#, на операційних системах відмінних від Microsoft Windows – Unix-подібних, Mac OS, та інших.

Варто зауважити, що розробкою Xamarin займаються ті самі люди, які займаються розробкою Mono. На даний момент (початок 2016 р.), Microsoft вкладає угоду із Xamarin, умови якої не розповсюджуються [3].

При розгляданні виконання додатку на iOS та Android можна виділити їх ключову різницю – реалізацію їх компіляції. Для виконання додатку в Android використовується віртуальна Java-машина Dalvik. Додатки, що пишуться на Java, компілюються в деякий проміжний байт-код, який інтерпретується за допомогою Dalvik у команди процесора вже у момент виконання програми (аналогічно тому, як працює CLR у .NET). Це так звана Just-in-time компіляція (компіляція на льоту). В iOS використовується інший механізм компіляції – Ahead-of-Time (компіляція перед виконанням). У Xamarin цю різницю враховано представленням окремих компіляторів для кожної із цих платформ, що дозволяють отримувати справжні для кожної із платформ додатки, які будуть виконуватись без використання браузеру, та матимуть можливість використовувати усі апаратні та програмні ресурси платформи [5].

Для iOS ситуація простіша – віртуальна машина відсутня, і програмний код повинен бути заздалегідь скомпільовано у машинний код. Для цієї мети використовується AOT компілятор Mono.

Для Android справедливо сказати, що при компіляції додатку відбувається переведення коду у проміжний байт-код, який зрозумілий віртуальній машині Mono. Віртуальна машина закладається у додаток. Mono і Dalvik написані за допомогою C, і працюють поверх ядра Linux (Android - на базі ядра Linux). При

запуску додатка на Android обидві віртуальні машини починають працювати пліч-о-пліч, і обмінюватись даними через спеціальний механізм wrapper.

Також, окрім особливостей розробки, слід вказати на складнощі та основні проблеми, що можуть виникнути при крос-платформній розробці:

- емулятори, як правило, мають обмежений функціонал, і у випадку з Android досить повільні. Тому варто мати пристрої, які працюють на IOS та Android;

- під керуванням Android працює цілий ряд різноманітних пристроїв, які абсолютно не схожі між собою, мають, подекуди принципово різні апаратні можливості. Особливо гостро це питання стосується тих смартфонів, які випускаються у Китаї;

- не можна сказати що проект, що розроблявся для Android, можливо зберегти та скомпілювати для IOS, - доведеться переписувати UI, та використовувати xCode. Але це не велика проблема, адже логіку та структуру коду це ніяк не зачіпає;

- у середньому можна використати близько 60% спільного коду для різних платформ. Хоча розробники Xamarin постійно підкреслюють тенденцію зростання портативності коду (у відсотковому відношенні) [6].

Таким чином, тема крос-платформні, особливо серед мобільних додатків є досить актуальною. А виділена нами можливість реалізації за допомогою Xamarin надає можливість якнайшвидшого вирішення поставленої проблеми з якнайменшими витратами ресурсів, без втрати швидкодії.

Список використаних джерел та літератури

1. Xamarin. За и против [Электронный ресурс] – Режим доступу: <https://habrahabr.ru/post/227863/>

2. Подробно о Xamarin [Электронный ресурс] – Режим доступу: <https://habrahabr.ru/post/188130/>

3. The Xamarin story [Электронный ресурс] – Режим доступу: <https://www.xamarin.com/about>

4. Android [Электронный ресурс] – Режим доступу:

<https://uk.wikipedia.org/wiki/Android>

5. Part 3 – Setting Up A Xamarin Cross Platform Solution [Электронный ресурс] – Режим доступа: https://developer.xamarin.com/guides/cross-platform/application_fundamentals/building_cross_platform_applications/part_3_-_setting_up_a_xamarin_cross_platform_solution/

6. Understanding the Xamarin Mobile Platform [Электронный ресурс] – Режим доступа: https://developer.xamarin.com/guides/cross-platform/application_fundamentals/building_cross_platform_applications/part_1_-_understanding_the_xamarin_mobile_platform/